ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Relating compound toxicity to molecular structure using machine learning

Master Thesis

Robin Bosshard

October 16, 2023

Advisors: Dr. Eliza Harris, Dr. K. Arturi, Lilian Gasser
Department of Computer Science, ETH Zürich

Abstract

Abstract goes here.

Contents

Chapter 1

Introduction

intro

Chapter 2

Literature Review

2.1 Background

2.2 Context

Chapter 3

Material and Methods

3.1 Invitrodb

The most recent release of the Toxicity Forecaster database, referred to as ToxCast's invitroDBv3.5, represents an extensive collection of high-throughput screening (HTS) targeted bioactivity data. This database encompasses information on a total of 9541 compounds, selectively screened across 2205 assay endpoints. The establishment of this resource owes its origins to the collaborative endeavors of two prominent institutions: the United States Environmental Protection Agency (EPA) through its ToxCast program and the National Institutes of Health (NIH) via the Tox21 initiative. Incorporating data collected from diverse research laboratories, this relational database is openly accessible to the public and can be downloaded directly from the official ToxCast website.

3.1.1 Presence matrix

Consider a collection of m assay endpoints, denoted by $A = \{a_1, a_2, \ldots, a_m\}$ and a set of n compounds represented as $C = \{c_1, c_2, \ldots, c_n\}$. To facilitate data comprehension, we introduce a *presence matrix* $P \in \{0,1\}^{m \times n}$. Rows, indexed by i, represent assay endpoints a_i, while columns, indexed by j, denote presence (1) or absence (0) of compound c_j in those endpoints. Matrix P is sparse due to the selective testing of compounds across different assay endpoints. A compound is considered present in an assay endpoint if it has undergone testing and a corresponding concentration-response series is available. See Figure 3.1 for a visual of the *presence matrix* P covering all assay endpoints and compounds in *invitroDBv3.5*.

A *concentration-response series* is represented as a set of k concentration-response pairs:

$$S = \{(conc_1, resp_1), (conc_2, resp_2), \ldots, (conc_k, resp_k)\}$$

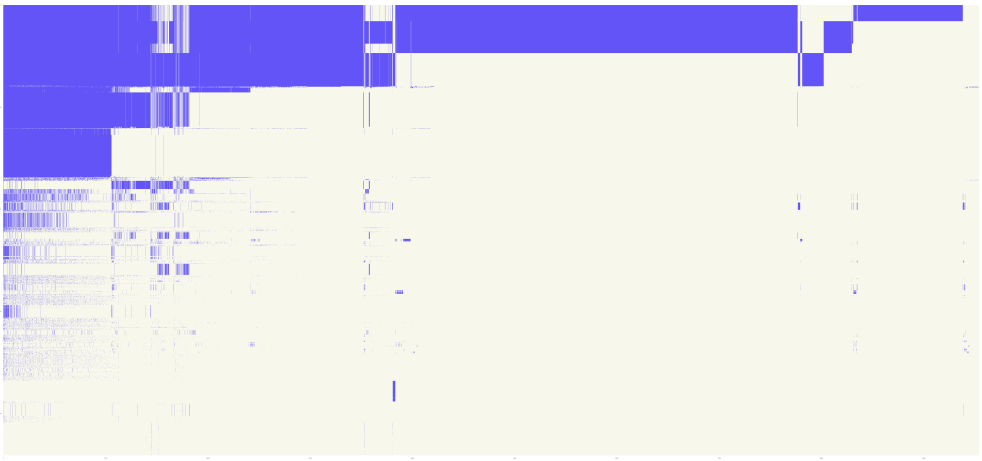

Figure 3.1: The *presence matrix* P covering all assay endpoints and compounds available in *invitroDBv3.5* with $m = 2205$ assay endpoints and $n = 9541$ compounds. The count, where $P_{ij} = 1$, indicates the availability of 3 342 377 concentration-response series for downstream analysis.

where k $conc_i$ values are not necessarily unique. The quantity of concentration-response pairs exhibits considerable variability among different compounds tested across various assay endpoints. In practice, concentrations are often subjected to multiple testing iterations, resulting in the formation of n_{conc} distinct concentration groups. Within each concentration group, the number of replicates is indicated by n_{rep}. Concentrations are transformed to the logarithmic scale using the unit μM (micromolar), while the responses are normalized to either fold-induction or percent-of-control units. Figure 3.2 showcases a concentration-response series for a compound tested within a single assay endpoint.

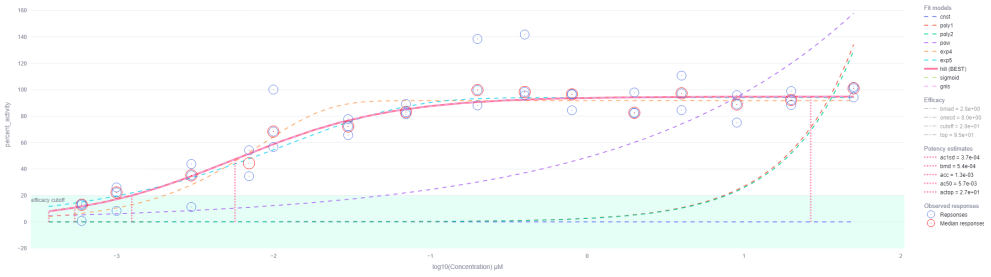

Figure 3.2: A concentration-response series for the compound *Estropipate* in the assay endpoint *TOX21_ERa_LUC_VM7_Agonist*. The series has a total of $k = 45$ concentration-response pairs and is composed of $n_{conc} = 15$ concentration groups, each with $n_{rep} = 3$ replicates.

3.2 Pytcpl

We introduce pytcpl, a streamlined Python package inspired by the R package tcpl, designed for processing high-throughput screening data. The package primarily focuses on providing essential features such as concentration-response curve fitting and allows for continuous hit-calling for compound bioactivity across diverse assay endpoints, akin to tcplfit2. Invitrodb version 3.5 release can optionally serve as backend database if desired. The package optimizes data storage and provides compressed raw data and metadata from *invitroDB* in Parquet files. This efficient strategy reduces storage needs, resulting in just 4 GB within the repository—compared to the original 80 GB database. This obviates the need for a cumbersome, large-scale database installation, rendering downstream analysis more accessible and efficient. Our package is crafted to accomodate cusomizable processing steps and facilitate interactive data visualization with curve surfer and empowers Python-oriented researchers to seamlessly engage in data analysis and exploration.

3.2.1 Preprocessing

First, all datapoints are collected from the database and assigned to the concentration response-series belonging to the respective ocompoud in the corresponding assay endpoint. The data is then filtered by the following criteria:

Data collection Compute efficacy cutoff Meet onditions for curve fitting

3.2.2 Curve Fitting

Introduce all candidate fit models

3.2.3 Hit Calling

Akaike criterion, 3 probabilities

3.2.4 Curve Surfer

Data visualization, overview of what is possible with the tool. Filter by assay endpoint, compound, etc.

3.3 Machine Learning Pipeline

3.3.1 Preprocessing

Subselecting the columns from the output tables generated by pytcpl: DTXSID identifier and continuous hitcall value. The feature inputs to the machine

learning model is a molecular structure represented as fingerprint generated from a SMILES string uniquely determined by the compounds DTXSID identifier. The SMILES string is a linear representation of a compound's molecular structure. The SMILES string is converted to a molecular graph, which is then converted to a feature vector. The feature vector is then used to train a machine learning model. The machine learning model is then used to predict the hitcall value for a given compound. The machine learning pipeline is illustrated in Figure **??**.

3.3.2 Binary Classification

The goal is to predict whether a compound is active or inactive for a given assay endpoint. We can formulate this as a binary classification problem, where the input is the compound's molecular structure fingerprint and the output is the hitcall value binarized by some decision threshold. The hitcall value is rendered to a binary variable, where 1 indicates that the compound is active and 0 indicates that the compound is inactive.

3.3.3 Regression

3.3.4 Massbank Validation

Chapter 4

Results and Discussion

sectionResults sectionEvaluation sectionDiscussion

Chapter 5

Conclusion

Appendix A

Appendix

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.